Faithful Completion of Images of Scenic Landmarks using Internet Images

Zhe Zhu, Hao-Zhi Huang, Zhi-Peng Tan, Kun Xu*, and Shi-Min Hu, *Member, IEEE*

Abstract—Previous works on image completion typically aim to produce visually plausible results rather than factually correct ones. In this paper, we propose an approach to *faithfully* complete the missing regions of an image. We assume that the input image is taken at a well-known landmark, so similar images taken at the same location can be easily found on the Internet. We first download thousands of images from the Internet using a text label provided by the user. Next, we apply two-step filtering to reduce them to a small set of candidate images for use as source images for completion. For each candidate image, a co-matching algorithm is used to find correspondences of both points and lines between the candidate image and the input image. These are used to find an optimal warp relating the two images. A completion result is obtained by blending the warped candidate image into the missing region of the input image. The completion results are ranked according to combination score, which considers both warping and blending energy, and the highest ranked ones are shown to the user. Experiments and results demonstrate that our method can faithfully complete images.

Index Terms—Image Generation, Image Completion, Image Matching, Image Blending.

1 Introduction

There is strong public demand to repair photographs, for example to remove an unwanted object or person from a wedding or travel photo, or to fill missing areas in an old, damaged photo. Image completion provides an effective tool for this purpose. It fills in missing or unwanted regions with new plausible content.

Image completion has been widely investigated in the computer graphics and image processing communities. One category of existing methods [1], [2], [3] fills missing regions using content from the same input image, based on texture and patch similarity. Recent Internet-based image completion work [4], [5] takes visual fidelity of image completion to a new level. Instead of searching within the input image, holes are filled by finding suitable image regions in a huge image database. While existing work is able to produce visually plausible results, they are usually not a *faithful* reconstruction of the real objects or scene that should have been there.

In this paper, we propose an image completion approach which aims to faithfully reconstruct a correct image. We assume that the input image was taken at a famous scenic landmark, and that a text label describing or naming the landmark is also given. We download thousands of images by searching the Internet using the text label, after which two-step filtering is applied to obtain a small set of candidate images from amongst the downloaded images. Specifically, we only retain candidates which are similar to the input image in gist feature space [6] and which are well registered with the input image. Thus, the candidate images generally contain the same scene as the input image, but taken with different camera parameters, from different viewpoints, or under different illumination conditions.

For each candidate image, in order to utilize it for completion, we need to align it with and warp it to the input image. To do so, we use a co-matching algorithm, which finds both point and line correspondences between the candidate image and the input image. Then, we adopt a mesh based warping strategy, and use a carefully designed energy function to preserve both point and line correspondences, and shapes. This energy is minimised iteratively. The completion result is obtained by blending the warped candidate image into the missing region of the input image. An optimal seam is found using graph-cut and blending is done in the gradient domain. Multiple completion images are generated, and the ones with the highest scores which consider both warping and blending energy are then returned to users. Experiments and results demonstrate that our method can effectively generate faithful completion images.

In summary, our work has three main contributions:

- A fully automatic approach to faithfully complete images of scenic landmarks. Previous approaches to this problem are not fully automatic [7] or cannot handle such complex cases [8] as our approach.
- A co-matching technique which, when matching line segments, utilizes geometric information from previously matched points. It is more efficient than the state-of-the-art line matching technique.
- A warping technique that can well register two images of non-planar scenes. The images may be taken from different viewpoints, under different illuminance conditions and with different lens distortions.

Z. Zhu, H.-Z. Huang, Z.-P. Tan, K. Xu and S.-M. Hu are with the TNList, Tsinghua University, Beijing 100084, China.
E-mail: ajex1988@gmail.com, huanghz08@gmail.com, tantantzp@qq.com, shimin@tsinghua.edu.cn.
**xukun@tsinghua.edu.cn, corresponding author.*

2 Related Work

Image Completion. Image completion approaches can be classified into diffusion based and example based approaches. Diffusion based methods [9], [10], [11] aim to extend image structures into small holes, and so cannot deal with large missing regions. Recent work has mainly focused on example based methods. Texture synthesis methods [12], [1] are example-based and perform well on images with holes in textured regions. The bottleneck lies in finding patch correspondences, which is time consuming. PatchMatch [2] utilizes the observation that neighbours of matched patches are also likely to be matched to significantly accelerate the process. Generalized PatchMatch [13] further improves PatchMatch by taking scaling and rotation of patches into consideration. Darabi et al. [14] improved the algorithm of Wexler et al. [1] in two ways: they calculate transformations of patches rather than only shifts of patches using the Generalized PatchMatch algorithm, and they add patch gradients in their distance metric in addition to colors. They also explored a similar multi-image completion process which fills a hole in one image with contents from other images. They adopted a *voting* strategy for visually plausible completion while we use a warping strategy for faithful completion. Furthermore, their candidate image is manually selected while ours is automatically selected. Sun et al. [15] proposed a structure-preserving image completion method which allows users to label structural information with lines. As well as line structures, other types of structures like symmetric structures [16] and planar structures [17] are also used to guide image completion. More recently, various automatic approaches have been proposed to analyse structural information both within images [18], [19] and between non-overlapping image pieces [20]. He et al. [18] compute the statistics of patch offsets automatically and formulate the completion process as a photomontage problem [21]. Huang et al. [19] detect planar surfaces and regularities within images and use them to guide patch search.

Some methods have considered faithful image completion [8][7] or faithful image expansion [22], i.e. showing visually correct content corresponding to the actual scene. They rely on images taken from similar viewpoints, and need to align these images to the viewpoint of the input image. A homography relates two images of the same scene taken from two different viewpoints if the scene is planar. Amirshahi et al. [8] use a single homography calculated by matching SIFT points to transform the candidate image to the input image: their method is thus limited to near-planar scenes. Whyte et al. [7] observed that non-planar scenes can be approximated by several planes, so they group homographies using matched SIFT points and transform the segmented planar regions with different homographies. Since segmenting an image into several planar regions is challenging, their method requires user intervention to obtain a correct segmentation, so is not automatic. Shan et al. [22] calculate structure from motion and reconstruct per-view depth-maps to warp each candidate image to the input image. As in [7], they have to solve a labeling problem to decide which warped image each pixel should come from. Such methods, which compose holes from multiple source images, all suffer from artifacts due to incompatibility of the warped sources. Our approach is superior, in that it gives a plausible warp relating two different view images in 2D, based on robust point and line matching.

Internet Image Processing. In recent years, researchers have developed many Internet-based techniques for e.g. scene completion [4], city reconstruction [23], photo enhancement [24] and image montage synthesis [25]. Such works construct a large database by downloading millions of images from the Internet, and use this as a data source for different image processing tasks. Our work also belongs to this category. A detailed survey can be found in [26].

Point Correspondences. Approaches for finding point correspondences in images can be classified as providing sparse correspondences [27], [28], [29], [30] or dense correspondences [31], [32]. Finding point correspondences is also referred to as keypoint matching. One of the most popular keypoint descriptors is SIFT [27], and in many cases, SIFT matches are used as initial correspondences. To filter out wrong matches, Cho et al. [28] cluster the initial matches based on their geometric distances and discard small clusters as outliers. In cases where a non-rigid mapping exists, an alternative approach is to model the correspondence problem as a graph matching problem [29]. Spectral techniques [30] and geometric blur descriptors [33], [34] have also been used for keypoint matching. Calculating dense correspondences is more challenge than calculating sparse correspondences. HaCohen et al. [31] proposed a patch based approach. They first calculate the nearest neighbour field by Generalized PatchMatch[13]; each patch is related to its nearest neighbour by a transformation. Two neighbourhood patches are regarded as consistent if the transformations to their nearest neighbours are similar. They link consistent patches to get dense correspondence. Another way to calculate dense correspondences is to use pixel based algorithms. Instead of using pixel intensities to calculate optical flow [35], Liu et al. [32] extract SIFT descriptors for each pixel thus extending frame-by-frame optical flow to scene-level image correspondences.

Line Correspondences. Various work has also been proposed for finding line correspondences [36], [37], [38], [39]. A straightforward approach is to use local features to describe line segments [37]. They calculate the histogram of gradient along a line segment, and use this as the descriptor for line segments. It is easy to extend this descriptor to curved lines. Another strategy is to use matched points to boost line matching [36]. In this work, Fan et al. observed that for two keypoints on the same side of a line, the ratio of their distances to the line is an affine invariant, which can be utilized to check if two lines match. A more reasonable way is to use epipolar geometry to register lines [39], but this requires camera calibration beforehand.

Image Warping. Warping is an efficient tool for image manipulation. Many works [40], [41], [42] generate control meshes in an image and define different forms of energy functions to optimize the positions of mesh vertices. These energy functions are application specific. To warp a panorama from an irrelugar shape to a rectangle, He et al. [40] consider shape preservation, straight line preservation and boundary constraints to optimize the guiding mesh. Carroll et al. [41] use shape preservation, straight line preservation and smoothness terms to constrain warping, thus making a wide-angle image look more friendly to human eyes. In video stabilization, Liu et al. [42] smooth the trajectory of a camera and project the original vertices into new positions. In their energy term, saliency and temporal coherence are also considered besides the commonly used similarity transformation term.

3 Overview

We take as input a scenic landmark image associated with a text label. Our approach consists of four steps; the pipeline of our approach is illustrated in Figure 1.

- **Candidate Image Downloading and Pre-Processing.** We download several thousand images by searching flickr.com using the associated text label. Following [4], we use gist scene descriptors [6] to perform initial filtering. The closest one hundred images to the input image according to gist scene descriptors are retained.
- **Candidate Images Filtering.** Next, the candidate images are reduced to a small set (typically 20) based on registration scores (Section 4). These images should contain the same scene as the input image, and can potentially produce good completion results. For each candidate image, we generate a completion image in the next step.
- **Co-Matching, Warping and Blending.** We first detect key points in both the candidate image and the input image, and find matches between the key points using agglomerative correspondence clustering [28]. A homography is computed for each cluster of matches. Line segments are also extracted in both images using [43]. To obtain better line matching results, we extend MSLD [37] by taking homographies recovered from point matching into consideration. Details are explained in Section 5. Next, we need to warp the candidate image to the input image. To do so, we adopt a mesh based warping strategy, using an energy function which preserves both point and line correspondences found in the co-matching step, as well as line structures and shapes. The energy function is minimized iteratively, as shown in Section 6. An example showing part of a candidate image before and after warping can be seen in Figures 2(a) and (b). To obtain a completion image, we need to blend the warped candidate image into the missing region of the input image. We first obtain an optimal seam using graph cut [44] (see Figure 2(c)), and then use the algorithm in [45] to blend the candidate image into the input image.
- **Ranking.** The above process provides multiple completion images. For each result, we compute a combination score, which considers both warping and blending energy. The completion images with the highest combination scores are returned to the user.

4 Candidate Image Filtering

To achieve faithful image completion, we need to use content from other images which contain the same scene as the input image. To do so, we first download several thousand images from Flickr using the text label associated with the input image. As most of the downloaded images are unrelated to the input image, we use gist scene descriptors [6] with 8 orientations and 4 scales to discard dissimilar images. The candidate images after initial filtering are denoted by S_g (their number is denoted N_g). Next, we need to find which candidate images contain the same content as the input. For this purpose, we define a registration score for each candidate image, which measures how well it can be registered to the input image. The N_p candidate images with highest registration scores are then retained and passed to further steps. The registration score P is defined as:

$$P(I_s,I_t) = \lambda_m \frac{P_m(I_s,I_t)}{\max(P_m(I_s,I_j))} - \lambda_a \frac{P_a(I_s,I_t)}{\max(P_a(I_s,I_j))}, \quad (1)$$

where I_j is from S_g, I_t denotes the candidate image, and I_s denotes the input image. The registration score involves two terms: the point matching term P_m and the affine registration term P_a. These terms are weighted by λ_m and λ_a, and are divided by their maximal values for normalization, respectively. The point matching term P_m is defined as:

$$P_m(I_s,I_t) = |C_{s,t}| \quad (2)$$

where $|C_{s,t}|$ denotes the number of point matches between the input image I_s and the candidate image I_t. Details of point matching are presented in Section 5. The affine registration term P_a is defined as:

$$P_a(I_s,I_t) = \frac{\sum_{(x_s,x_t)\in C_{s,t}} \|A_t x_t - x_s\|}{B} \quad (3)$$

where B denotes the size of a bounding box that contains all matching points, and A_t is the optimal affine transformation matrix that matches all points from the input image to the candidate image, which is obtained using the algorithm in [46].

In our implementation, we empirically set the weights $\lambda_m = 2$ and $\lambda_a = 1$. The number of retained images after initial filtering is set to $N_g = 100$, and the number of final retained candidate images is set to $N_p = 20$. Our experiments show that this setting gives a good balance between speed and accuracy. Our Matlab implementation takes about 400 seconds for this step (not including the time to download images).

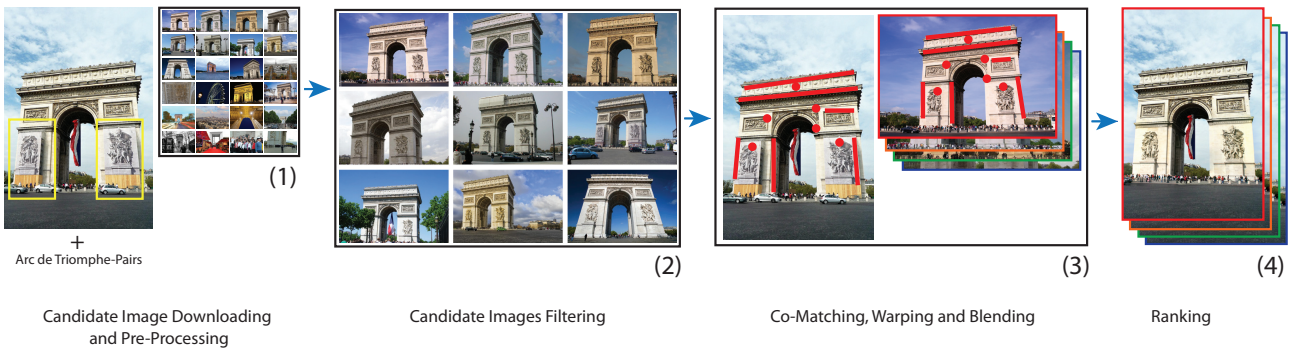


Fig. 1. Pipeline. (1) The input is a source image of a famous landmark, plus a keyword describing it. Image search engines are used to download initial candidate images. Gist descriptors are used to filter the initial image set. (2) We select potential candidates by further filtering. (3) For each candidate image, we match points and line segments, and compute an optimized warp from the candidate image to the source image. We then cut the appropriate part from the candidate image and blend it into the source image. (4) A ranking function is used to select the best completion results.

(a) (b) (c)

Fig. 2. (a) Part of the candidate image. (b) Warped result.(c) Optimal seam (blue).

5 Co-Matching

The previous section explained how to obtain a small set of candidate images. These candidate images generally contain the same scene as the input image, but taken with different camera parameters, at different viewpoints or under different illumination conditions. Now, we need to find correspondences between each candidate image and the input image, in order to align them later. Various work has been proposed for finding point correspondences [27] and line correspondences [36] between two images. In order to obtain more robust matching results, we have developed a co-matching algorithm, which finds both point and line matches. The main idea is to use the recovered homography based on point correspondences to further improve the accuracy of line correspondences. The co-matching algorithm works as follows:

Point Correspondences. We first detect MSER region-s [47] and extract SIFT [27] features to match the key points of the two images. Next, we cluster the matched points based on the feature distance between each matched point pair. We adopt agglomerative correspondence clustering [28] to cluster the matched pairs and discard small clusters (with less than 20 pairs) as outliers. Then, we compute a homography by least-squares fitting for each remaining cluster. A recovered homography is illustrated in Figure 4.

Line Correspondences. First, we use the method of [43] to detect line segments in both images (see Figure 3). To match line segments between two images, we need a distance metric for line segments. We use:

$$D(l_i, l_j) = D_{\text{feat}}(l_i, l_j) + \alpha D_{\text{geo}}(l_i, l_j), \quad (4)$$

where l_i, l_j are line segments in the two images respectively. Our distance metric takes two terms into account: the feature distance D_{feat} and the geometric distance D_{geo}. α is a weight to control the relative contributions of the two terms, set to $\alpha = 2$ empirically. The feature distance D_{feat} is computed as the difference of the MSLD descriptors [37] of the two line segments. The geometric distance measures how well the two line segments match each other under the homography just recovered, and is defined as:

$$D_{\text{geo}}(l_i, l_j) = \frac{1}{2}(D_{\text{Hough}}(H_i l_i, l_j) + D_{\text{Hough}}(H_j^{-1} l_j, l_i)), \quad (5)$$

where D_{Hough} represents the distance of two lines in Hough space; H_i and H_j are homographies at l_i and l_j, respectively, which have been recovered in the previous point matching step. Both feature and geometric distances are normalized to $[0,1]$ by dividing by their maximal values, respectively.

We then use the line distance metric in Equation 4 to compute all-pair distances between line segments in the

Fig. 3. Line detection. Blue lines show detected line segments whose length is at least 40 pixels.

Fig. 4. Homography region partition. Matched points in the same cluster are marked in the same color. For each cluster, we fit a homography based on the matched points.

two images. Two line segments l_i and l_j are considered as matched if l_i is closest to l_j among all line segments in the input image, and l_j is also closest to l_i among all line segments in the candidate image.

Evaluation. To evaluate the effectiveness of our co-matching algorithm, we compared it to two state-of-the-art line matching algorithms: Fan's algorithm [36] and Wang's algorithm [37]. To enable a fair comparison, all the algorithms should have a similar number of matched lines. In some cases we should increase the number of matched lines for a certain algorithm while decrease the number of matched lines for another algorithm. To increase the number of matched lines, we loosened the matching criteria for the algorithm: two line segments from two images are matched if they belong to each other's top n matches where n is a parameter. To decrease the number of matched lines, we set a distance threshold t to discard some matched line pairs whose distances are larger than t. We adjusted n and t to let these algorithms have a similar number of matched lines. The lines were generated using the method in [43], and used as input for all 3 algorithms. As shown in Table 1, our line matching algorithm has a higher correct matching rate than these previous methods.

6 Warping Constrained by Points and Lines

After finding point and line matches between the input image and the candidate image, the next tasks are to align the matched points and lines, and to warp the candidate image to the input image. To do so, we adopt a mesh-based warping algorithm with a carefully chosen energy function. We parameterize the candidate image using a uniform mesh with 20×20 vertices. The energy function considers constraints on both point and line correspondences, as well as preserving shape and line structures. In the following, we explain the details of the energy function, and how we use the mesh grid for warping.

6.1 Energy Function

Our energy function considers constraints on matched points and lines, as well as preserving structures like shapes and other straight lines. We denote the initial mesh by $\hat{V}$, and the output optimized mesh as V. The vertices of the grid V are denoted by its $x-$ and $y-$ coordinates: $V = \{(x_i, y_i), \quad 1 \leq i \leq N\}$, where N is the number of vertices. $\hat{V}$ is defined in a similar way.

Point Constraints. The point constraint term constrains the matched points to remain close after warping. Specifically, the point constraint term E_P is defined as:

$$E_P(V) = \sum_{i=1}^{K} (x_{i,s} - x_{i,r})^2 + \sum_{i=1}^{K} (y_{i,s} - y_{i,r})^2, \qquad (6)$$

where K is the number of matched point pairs; $(x_{i,s}, y_{i,s})$ and $(x_{i,r}, y_{i,r})$ are the positions of the points of the $i-$th matched pair (subscripts s and r represent the input and warped candidate images, respectively). Note that the position of each point in the warped candidate image is found by bilinear interpolation of its four adjacent vertexes, so that E_P can be written as a quadratic function of vertex positions V.

Line Constraints. Since human vision is very sensitive to artifacts along straight lines, we also wish to preserve line correspondences during warping. We constrain the lines in the warped candidate image to have similar orientations and positions to the corresponding lines in the input image, and to remain straight. We use two terms E_{LC1} and E_{LC2} to impose line constraints, which preserve orientation and position of lines, respectively. Denote the lines in the input and the warped candidate image as $l_{i,s}$ and $l_{i,r}$, respectively. Inspired by [48], we cut a line segment into smaller segments if it crosses mesh edges.

For each line segment in a mesh quad, its orientation vector is computed as the direction from one endpoint to the other. For each matched pair of line segments, we denote the orientation vector of the line segment in the input and warped candidate images as $\hat{e}$ and e, respectively. The first term E_{LC1} is defined as the mean distortion over all segments:

$$E_{LC1}(V) = \frac{1}{N_{LC}} \sum_{i=1}^{N_{LC}} \|s_i R_i \hat{e}_i - e_i\|^2, \qquad (7)$$

where N_{LC} is the number of segments, $R_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}$ is a rotation matrix, and s is a scaling factor. Unlike in [48], here the rotation angle θ_i is given by the matched line pair with index i. Minimizing this energy term with respect to s gives: $s = (\hat{e}^T \hat{e})^{-1} \hat{e}^T R^T e$. Since e can be represented by a linear function of V, E_{LC1} can also be written as a quadratic function of V.

TABLE 1
Line matching: our results compared to Fan's algorithm and Wang's algorithm.

		total matched lines	correct match	correct rate	time(s)	source image	candidate image
Kensington Castle	Fan's method	108	102	94%	16		
	Wang's method	106	82	77%	5		
	Our method	103	101	98%	16	1000*667	1504*1000
Notre Dame	Fan's method	135	108	80%	29		
	Wang's method	125	107	85%	4		
	Our method	140	139	99%	15	1000*928	922*1229
Duomo	Fan's method	198	182	92%	22		
	Wang's method	194	168	87%	4		
	Our method	202	188	93%	8	1000*664	1000*709
Palazzo Santa Sofia	Fan's method	266	256	96%	32		
	Wang's method	278	255	91%	6		
	Our method	243	235	97%	15	1024*632	1024*768
Rialto Bridge	Fan's method	164	159	97%	5		
	Wang's method	133	114	86%	3		
	Our method	152	148	97%	5	820*403	820*403
Leaning Tower of Pisa	Fan's method	40	35	88%	6		
	Wang's method	32	27	84%	2		
	Our method	35	32	91%	6	684*1024	683*1024
Arc de Triomphe -Paris	Fan's method	138	137	99%	7		
	Wang's method	162	139	86%	4		
	Our method	146	146	100%	6	533*800	968*667
Big Ben	Fan's method	44	30	68%	4		
	Wang's method	43	32	74%	3		
	Our method	45	41	91%	4	1024*683	685*1024

The second term E_{LC2} is designed to constrain the position of lines. Inspired by [48], we transform a line $y = ax + b$ in image space into a point (r,θ) in Hough space. For an arbitrary point (x_0, y_0) on this line, $x_0\cos\theta + y_0\cos\theta = r$. We constrain the range of θ to be $[0, 2\pi)$, so that r will always be positive. For each segment in a quad, we denote its final position after warping as (r_i, θ_i) in Hough space. Meanwhile, we also know its ideal position $(\hat{r}_i, \hat{\theta}_i)$ according to the corresponding line in the source image. To simplify the problem, we suppose that θ_i is always equal to $\hat{\theta}_i$, so that r_i is the only unknown and can be represented by a linear combination of the four quad vertexes. This assumption is reasonable provided we give a large weight to E_{LC1}. Now the second line constraint term E_{LC2} can be defined as:

$$E_{LC2}(V) = \frac{1}{N_{LC}} \sum_{i=1}^{N_{LC}} \|\hat{r}_i - r_i\|^2. \tag{8}$$

E_{LC2} can also be rewritten as a quadratic function of V.

Line Preservation. Besides lines in correspondence, we also want to constrain all unmatched lines to be straight after warping. We define the line preservation term E_{LP} as:

$$E_{LP}(V, \{\theta_m\}) = \frac{1}{N_{LP}} \sum_{i=1}^{N_{LP}} \|s_i R_i \hat{e}_i - e_i\|^2. \tag{9}$$

Unlike the line constraint term, here the rotation angle for each segment is also an unknown. Following [48], we quantize the orientation range into 50 bins. Each segment can be assigned to a bin according to its original orientation. We encourage segments in the same bin m to share the same rotation angle θ_m. Actually, some of the θ_m can be pre-computed. For a line with correspondence, we already know its rotation angle θ. Then we can compute which bin it belongs to and fix the θ_m for that bin as θ. Meanwhile, the rotation angles of lines without correspondences will be guided by those with correspondences.

Shape Preservation. In order to preserve the shape of the image content, we define a shape preservation energy term as:

$$E_S(V) = \frac{1}{N_q} \sum_{q=1}^{N_q} \|(A_q(A_q^T A_q)^{-1} A_q^T - I)V_q\|^2, \tag{10}$$

where N_q is the number of quads in the mesh, A_q is defined as:

$$A_q = \begin{bmatrix} \hat{x_0} & -\hat{y_0} & 1 & 0 \\ \hat{y_0} & \hat{x_0} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{x_3} & -\hat{y_3} & 1 & 0 \\ \hat{y_3} & \hat{x_3} & 0 & 1 \end{bmatrix}, \tag{11}$$

and V_q is defined as:

$$V_q = [x_0, y_0, x_1, y_1, ..., x_3, y_3]^T. \tag{12}$$

Here we denote the four corners of the quad as $(x_0, y_0), \ldots, (x_3, y_3)$. E_S encourages each quad to undergo a similarity transformation; a detailed derivation can be found in [49].

Total Energy Function. Finally, the total energy function E is defined as:

$$E(V, \{\theta_m\}) = E_P + \lambda_{LC1}E_{LC1} + \lambda_{LC2}E_{LC2} + \lambda_{LP}E_{LP} + \lambda_S E_S, \tag{13}$$

where each λ controls the weight of the corresponding term. Same as previous works [40], [41] we give high weight for line orientation term. In our implementation, we empirically set λ_{LC1} to 100000, λ_{LC2} to 1, λ_{LP} to 100, λ_S to 1000.

6.2 Iterative Two-step Optimization

Minimizing the energy to find V and $\{\theta_m\}$ in Equation 13 at the same time is difficult. Instead, we solve this problem using an iterative two-step approach. In the first step, we fix $\{\theta_m\}$ and solve for V, making E a quadratic function which can be solved by a linear least-squares solver. Since we have only 400 vertexes in V (i.e. 800 unknowns), the linear system can be solved rapidly. In the second step, we fix V and solve for $\{\theta_m\}$. To simplify the problem, we use an intuitive averaging strategy instead of a non-linear solver. Specifically, each θ_m can be estimated using the average of the relative angle between e and $\hat{e}$ for all segments in the bin indexed m. However, not all elements in $\{\theta_m\}$ should be updated. As mentioned in Section 6.1, the rotation angle θ for a line with correspondence is already known and the related bin in $\{\theta_m\}$ should be fixed to θ. We iteratively apply the above two steps, and usually 5 iterations suffice to obtain a satisfactory result.

7 Result Ranking

After obtaining one completion result for each candidate image, we now need to select the best results from those obtained. We define the score of a completion result as:

$$R = \lambda_r P(I_s, I_t) + \lambda_w P_w(I_s, I_t) + \lambda_g P_g(I_s, I_t) \qquad (14)$$

where I_s,I_t denote the input and the corresponding candidate images, respectively; $P(I_s, I_t)$ is the registration score defined in Section 4; P_w is the warping score and it is defined as:

$$P_w(I_s, I_t) = 1 - \frac{E_w(I_s, I_t)}{\max(E_w(I_s, I_j)), j \in S_g} \qquad (15)$$

where $E_w(\cdot,\cdot)$ is the total warping energy of the two images defined in Eqn. 13. P_g is the normalized energy of the graph-cut boundary segmentation:

$$P_g(I_s, I_t) = 1 - \frac{E_g(I_s, I_t)}{\max(E_g(I_s, I_j)), j \in S_g} \qquad (16)$$

where $E_g(\cdot,\cdot)$ is the graph-cut energy used in [21]. We add this term because a poorly segmented boundary always leads to obvious artifacts.

In our implementation, we empirically set the weights in the ranking function to $\lambda_r = 0.2$, $\lambda_w = 0.4$ and $\lambda_g = 0.4$, respectively. By default, our system returns the result with the highest score. Users can also select from further results if dissatisfied with the top ranked result. Various ranked results are illustrated in Figure 5.

8 Results and Discussion

8.1 Implementation Details.

We have implemented our method using Matlab on a PC with an Intel Core i7 3.4GHz CPU and 8GB memory. Our system takes about 2 minutes to obtain the final completion results for a typical input image with 1200×800 resolution (the time for downloading images is not included).

8.2 Demonstration

Figure 6 demonstrates the effectiveness of our system, showing completion results for various scenes. In the first row, the input image (i.e. the left column) is taken at the Rialto Bridge in Venice. Notice that there is an unwanted flag in the middle of the bridge, occluding part of it. Our method successfully removed the flag and replaced it with content from other images. Our result is given in the right column; candidate images are shown in the middle column. Other rows show results at Kensington Palace (with an unwanted statue in front of the palace), at a bridge in a traditional Chinese town (with unwanted people), at a Buddhist temple in Japan (the bottom-left corner is occluded by a tree), and at the Leaning Tower of Pisa (one of the floors is covered by a fence). Our method can handle all these cases well.

8.3 Need for Both Point and Line Constraints

The reason we do co-matching is that both point constraints and line constraints play important roles in our warping. We illustrate this in Figure 7. In Figure 7(a), we want to remove the man in front of the Duomo of Milan. Without point constraints, the entire wall cannot be well registered: see Figure 7(b). When we add point constraints but not line constraints, although on the whole the result is better registered than in Figure 7(b), there is still severe misalignment in the line structures: see Figure 7(c). Figure 7(d) is the result using both point and line constraints. It is well registered and the completion result does not suffer from artefacts.

8.4 Comparison with PatchNet and Scene Completion

We next compare our approach with state-of-the-art Internet based image completion methods including a scene completion method [4] and PatchNet [5]. To enable a fair comparison, we built an image library containing hundreds of thousands of images, and used same image library for all three methods. As shown in Figure 8, for all four examples, our method is able to generate faithful completion results, while the scene completion method [4] and PatchNet [5] both generate unfaithful (although visually plausible) results. This is unsurprising, since these two methods only find similar images (or similar content in images) and aim to generate *plausible* results. In comparison, our method uses images taken at the same scene for completion, and hence is more effective in generating *faithful* completion results.

8.5 Comparison with Moving Least-Squares Deformation

We now compare our warping method in Section 6 to moving least-squares deformation (MLSD) [50]. The candidate

Fig. 5. Ranked results, showing 4 representatives. Rank 1 is the best result; Rank 4 suffers from a non-smooth graph-cut boundary; Rank 9 has many fewer matching points; Rank 15 has too large a viewpoint difference.

image used was the same in both cases. MLSD uses point constraints as well as line constraints. With MLSD, too many point constraints lead to extreme distortion, so we picked 20 matched pairs with the highest confidence in our results. For similar reasons, we picked the 5 best matched lines. After deformation, we blended the deformed parts into the source image. In the first row of Figure 9, the clock of Big Ben was replaced. Neither point constraints nor line constraints can preserve local shapes. Compared with the results in the first row, the distortion in the deformation using MLSD is more obvious in the second row of Figure 9, where we tried to remove the boat in front of the Palazzo Santa Sofia.

8.6 Comparison with Single and Multiple Homography Approaches

Our approach is superior to the single homography approach in [8] and the multiple homography approach in [7]. A single homography cannot register the source image and the candidate image for non-planar scenes. Multiple homography suffer from incompatibility between the images due to different white balance, different resolution and different lighting conditions. In Figure 10, we compare our approach with single and multiple homography approaches. For the single homography approach, we use the same candidate image as in our approach. For the multiple homography approach, we use the top 10 images after initial filtering. In the first row, we remove the statue in front of St. Basil's Cathedral. Using the single homography approach, the left of the image is excessively stretched. The result of multiple homography approach seems good at first glance, but the blue and white top is doubled. In the second row, we remove the part of the central television building that is being built. The single homography approach fails to register the building well while this problem is less severe for the multiple homography approach. However, different parts coming from from different candidate images leads to color inconsistency, which is the main artefact in this case. In the third row, we recover the occluded part of the Royal Albert Hall. The single homography approach again suffers from misalignment while for the multiple homography approach the bottom right statue is missing. In all the above cases, our approach generates faithful completion results. We give more comparisons in the supplemental material.

8.7 Depth of Field Extension

Our approach can also be used to change the depth of field of a photograph, producing interesting effects. In Figure 11, we replace the blurred objects with sharp ones to increase the depth of field.

8.8 Limitations

Our method may fail in some cases. For example, if the viewpoint differs greatly between the input and candidate images, it may be hard to find a reasonable warp from the candidate image to the input image, leading to results with inconsistent alignment (see Figure 12, first row). Furthermore, if there are large tone and illumination differences between the input image and the candidate image, visible color inconsistency artifacts may result (see Figure 12, second row).

Another limitation of our approach is that it cannot handle cases when the target region is too large. This is because our approach relies on correspondences between two images outside the target region, and if the target region is too large, there will be few matches. The point and line terms of the energy function will have less impact so the optimization cannot give a reasonable warp (see Figure 12, third row).

Fig. 6. Further completion results. Left: image to be completed; yellow rectangles mark the regions to be completed. Middle: candidate images. Right: completion result.

(a) Source image (b) Result without point constrains (c) Result without line constrains (d) Result with both constrains

Fig. 7. The necessity of using both point constrains and line constrains in the warping.

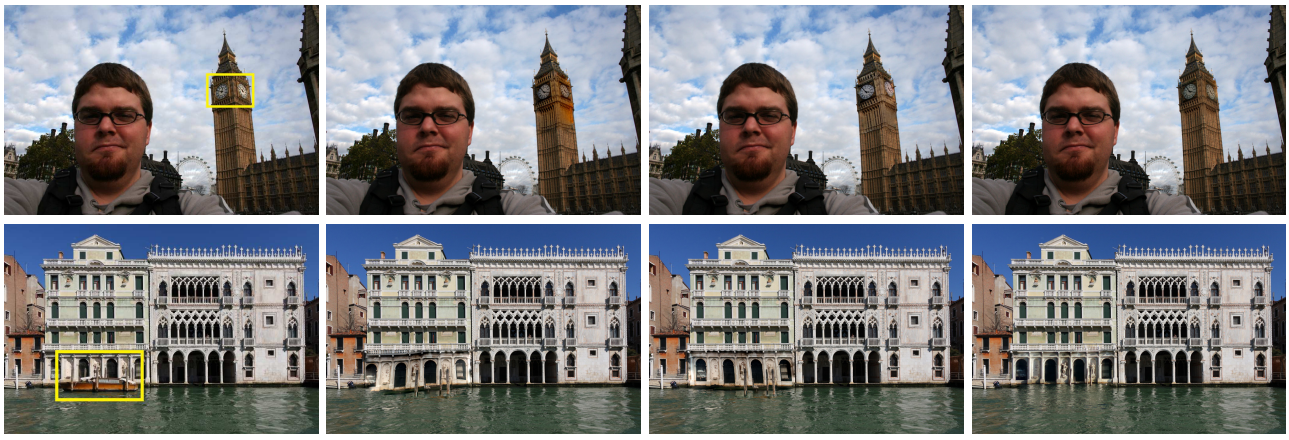
Source image Scene completion result PatchNet result Our completion result

Fig. 8. Comparison to scene completion [4] and PatchNet [5].

9 Conclusions and Future Work

We have proposed an approach for faithful completion of scenic landmark images using Internet images. The input comprises an image, a user given text label naming the landmark, and a region mask indicating where the image is to be completed. The completion process is fully automatic. Our method first downloads thousands of images from the Internet using the provided text label, and reduces them to a small set of candidate images through two-step filtering. For each candidate image, we apply co-matching to find point and line matches between the input image and the candidate, and compute a warp relating it to the input image. A completion result is obtained through gradient domain blending. The completion results with highest combination scores, which consider both warping and blending energy, are then displayed to users. We have validated our approach on many famous landmarks; experiments show that our approach can generate faithful results in most cases.

In future, to improve result quality, we plan to add an additional color transfer step to deal with cases when there are large tone and illumination differences between the input and candidate images. We also plan to extend our warping method to more general cases, such as adding support for panoramas.



Source image | With point correspondence | With line correspondence | Our completion result

Fig. 9. Comparison with moving least squares deformation (MLSD) [50].

Source image | Single homography | Multiple homography | Our completion result

Fig. 10. Comparison with single homography [8] and multiple homography [7] approaches.

ACKNOWLEDGMENT

This work was supported by the National High Technology Research and Development Program of China (Project Number 2013AA013903) and the Natural Science Foundation of China (Project Number 61272226/61170153/61120106007), Research Grant of Beijing Higher Institution Engineering Research Center, and Tsinghua University Initiative Scientific Research Program.

REFERENCES

[1] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, 2007.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24–34, 2009.

[3] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Transactions on Image Processing*, vol. 13, pp. 1200–1212, January 2004. MSR-TR-2003-83.

[4] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, 2007.

[5] S.-M. Hu, F.-L. Zhang, M. Wang, R. R. Martin, and J. Wang, "Patchnet: A patch-based image representation for interactive library-driven image editing," *ACM Trans. Graph.*, vol. 32, pp. 196:1–196:12, Nov. 2013.

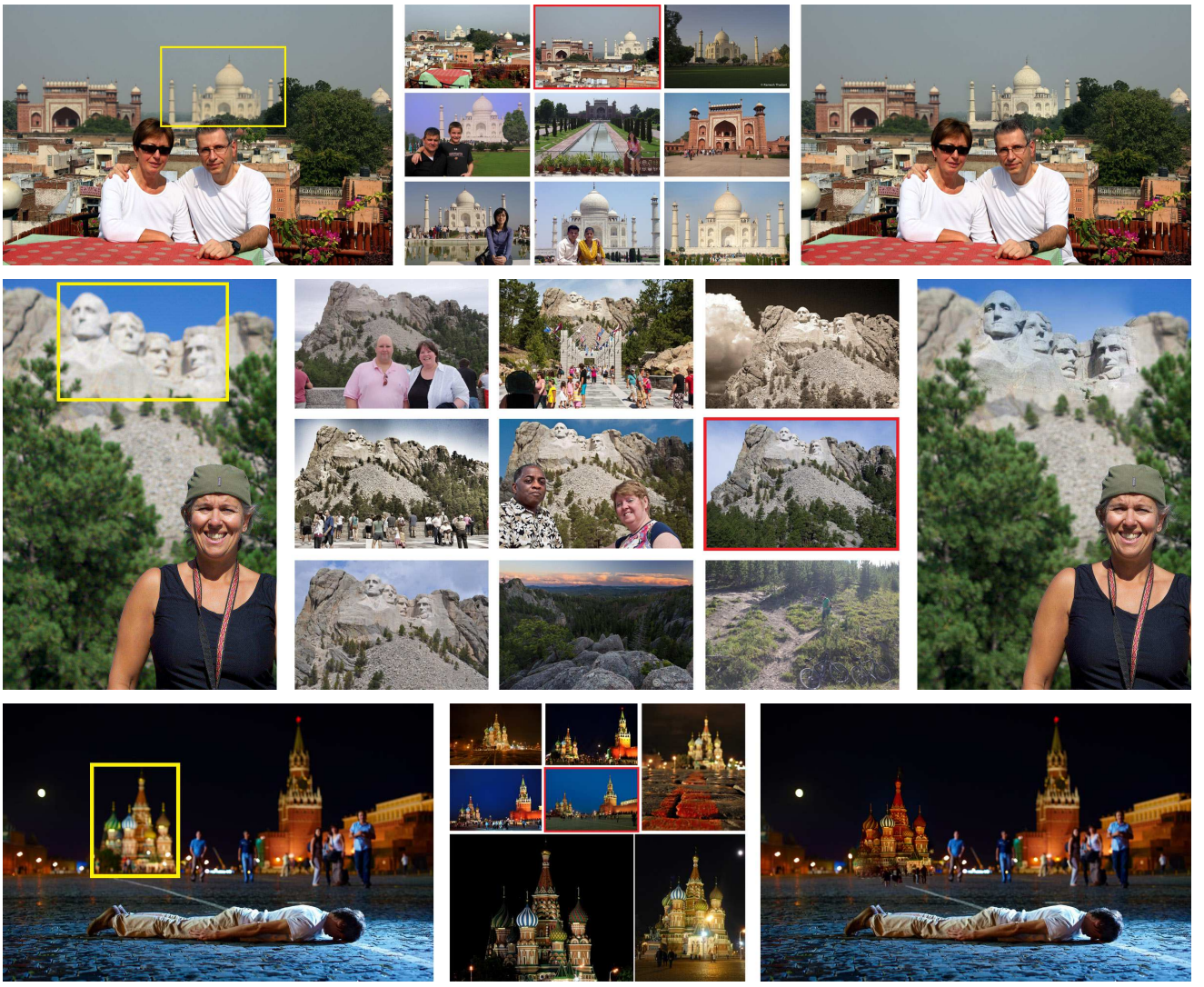


Fig. 11. Increasing the depth of field.

[6] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, (Washington, DC, USA), pp. 273–, IEEE Computer Society, 2003.

[7] O. Whyte, J. Sivic, and A. Zisserman, "Get out of my picture! internet-based inpainting," in *Proceedings of the 20th British Machine Vision Conference, London*, 2009.

[8] H. Amirshahi, S. Kondo, K. Ito, and T. Aoki, "An image completion algorithm using occlusion-free images from internet photo sharing sites," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E91-A, pp. 2918–2927, Oct. 2008.

[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, (New York, NY, USA), pp. 417–424, ACM Press/Addison-Wesley Publishing Co., 2000.

[10] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *Image Processing, IEEE Transactions on*, vol. 10, pp. 1200 –1211, aug 2001.

[11] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II – 707–12 vol.2, june 2003.

[12] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, (New York, NY, USA), pp. 341–346, ACM, 2001.

[13] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *European Conference on Computer Vision*, Sept. 2010.

[14] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, "Image Melding: Combining Inconsistent Images using Patch-based Synthesis," *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2012)*, vol. 31, no. 4, 2012.

[15] J. Sun, L. Yuan, J. Jia, and H. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, 2005.

[16] J.-. B. Huang, J. Kopf, N. Ahuja, and S. B. Kang, "Transformation guided image completion," in *International Conference on Computational Photography*, April 2013.

[17] D. Pavić, V. Schönefeld, and L. Kobbelt, "Interactive image completion with perspective correction," *The Visual Computer*, vol. 22, no. 9-11, pp. 671–681, 2006.

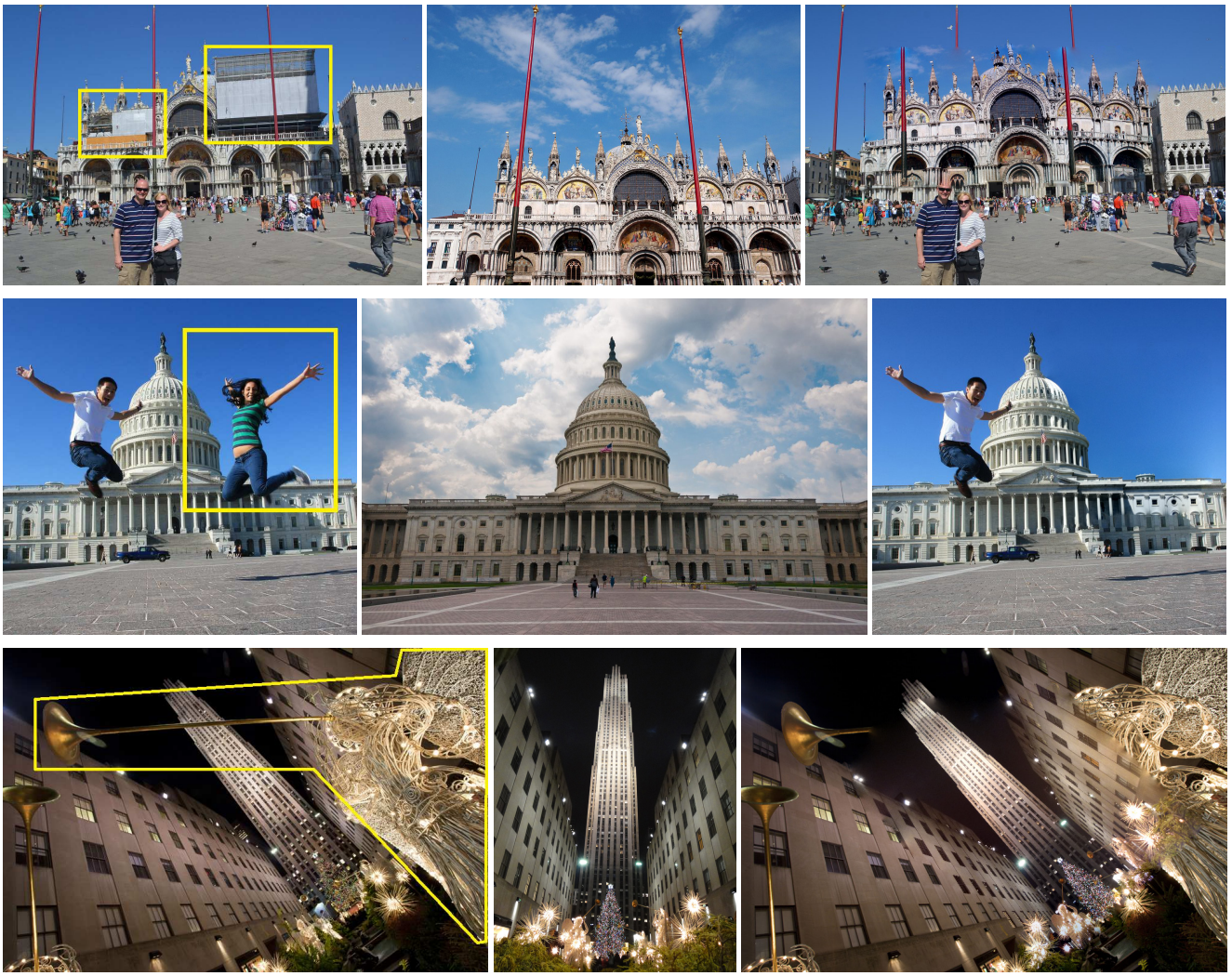


Fig. 12. Failures. Left: input image. Middle: candidate image. Right: completion result.

[18] K. He and J. Sun, "Statistics of patch offsets for image completion," in *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, ECCV'12, (Berlin, Heidelberg), pp. 16–29, Springer-Verlag, 2012.

[19] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, pp. 129:1–129:10, July 2014.

[20] H. Huang, K.Yin, M. Gong, D. Lischinski, D. Cohen-Or, U. Ascher, and B. Chen, "mind the gap: Tele-registration for structure-driven image completion," *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA 2013)*, vol. 32, pp. 174:1–174:10, 2013.

[21] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Trans. Graph.*, vol. 23, pp. 294–302, Aug. 2004.

[22] Q. Shan, C. Brian, F. Yasutaka, H. Carlos, and M. S. Steven, "Photo uncrop.," in *Computer Vision - ECCV 2014*, 2014.

[23] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Commun. ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[24] C. Zhang, J. Gao, O. Wang, P. Georgel, R. Yang, J. Davis, J.-M. Frahm, and M. Pollefeys, "Personal photograph enhancement using internet photo collections," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, pp. 262–275, Feb 2014.

[25] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2photo: Internet image montage," *ACM Trans. Graph.*, vol. 28, pp. 124:1–124:10, Dec. 2009.

[26] S.-M. Hu, T. Chen, K. Xu, M.-M. Cheng, and R. R. Martin, "Internet visual media processing: a survey with graphics and vision applications.," *The Visual Computer*, vol. 29, no. 5, pp. 393–405, 2013.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[28] M. Cho, J. Lee, and K. M. Lee, "Feature correspondence and deformable object matching via agglomerative correspondence clustering.," in *ICCV* [51], pp. 1280–1287.

[29] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization.," in *ECCV (2)* (D. A. Forsyth, P. H. S. Torr, and A. Zisserman, eds.), vol. 5303 of *Lecture Notes in Computer Science*, pp. 596–609, Springer, 2008.

[30] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, (Washington, DC, USA), pp. 1482–1489, IEEE Computer Society, 2005.

[31] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Trans. Graph.*, vol. 30, pp. 70:1–70:10, July 2011.

[32] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[33] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondence," in *In CVPR*, pp. 26–33, 2005.

[34] A. C. Berg and J. Malik, "Geometric blur for template matching," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conf. on*, vol. 1, pp. I—-607—-I—-614 vol.1, 2001.

[35] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, (San Francisco, CA, USA), pp. 674–679, Morgan Kaufmann Publishers Inc., 1981.

[36] B. Fan, F. Wu, and Z. Hu, "Line matching leveraged by point correspondences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 390–397, IEEE, June 2010.

[37] Z. Wang, F. Wu, and Z. Hu, "Msld: A robust descriptor for line matching.," *Pattern Recognition*, vol. 42, no. 5, pp. 941–953, 2009.

[38] A. Bartoli, M. Coquerelle, and P. F. Sturm, "A framework for pencil-of-points structure-from-motion.," in *ECCV (2)* (T. Pajdla and J. Matas, eds.), vol. 3022 of *Lecture Notes in Computer Science*, pp. 28–40, Springer, 2004.

[39] C. Schmid and A. Zisserman, "Automatic Line Matching across Views," in *International Conference on Computer Vision & Pattern Recognition*, (San Juan, Porto Rico), pp. 666–671, IEEE Computer society, 1997.

[40] K. He, H. Chang, and J. Sun, "Rectangling panoramic images via warping," *ACM Trans. Graph.*, vol. 32, pp. 79:1–79:10, July 2013.

[41] R. Carroll, M. Agrawal, and A. Agarwala, "Optimizing content-preserving projections for wide-angle images," in *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, (New York, NY, USA), pp. 43:1–43:9, ACM, 2009.

[42] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Trans. Graph.*, vol. 28, pp. 44:1–44:9, July 2009.

[43] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[44] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Transactions on Graphics, SIGGRAPH 2003*, vol. 22, pp. 277–286, July 2003.

[45] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski, "Coordinates for instant image cloning," *ACM Trans. Graph.*, vol. 28, pp. 67:1–67:9, July 2009.

[46] J. Ho, A. M. Peter, A. Rangarajan, and M.-H. Yang, "An algebraic approach to affine registration of point sets.," in *ICCV* [51], pp. 1335–1340.

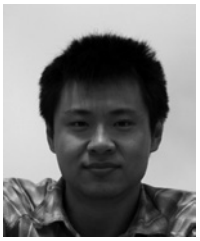
[47] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference*, pp. 36.1–36.10, BMVA Press, 2002. doi:10.5244/C.16.36.

[48] C.-H. Chang and Y.-Y. Chuang, "A line-structure-preserving approach to image resizing.," in *CVPR*, pp. 1075–1082, IEEE, 2012.

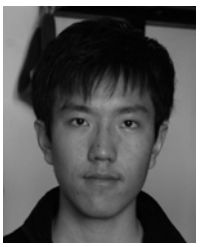
[49] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin, "A shape-preserving approach to image resizing.," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1897–1906, 2009.

[50] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *ACM Trans. Graph.*, vol. 25, pp. 533–540, July 2006.

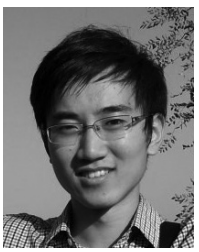
[51] *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, IEEE, 2009.



Zhe Zhu is a PhD student in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor's degree in Wuhan University in 2011. His research interests in Computer Vision and Computer Graphics.



Haozhi Huang is a PhD student in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor's degree in the same university in 2012.



Zhipeng Tan is an undergraduate student in Department of Computer Science and Technology, Tsinghua University.

Kun Xu is an associate professor in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor and doctor's degrees from the same university in 2005 and 2009, respectively. His research interests include realistic rendering and image/video editing.

Shimin Hu received the PhD degree from Zhejiang University in 1996. He is currently a professor in the department of Computer Science and Technology, Tsinghua University, Beijing. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is associate Editor-in-Chief of The Visual Computer, associate Editor of IEEE Transactions on Visualization and Computer Graphics, Computer & Graphics and Computer Aided Design.